

```

struct monotonic_queue {
    int A[MX], B[MX], C[MX];    // normal , for min , for max
    int head_a, tail_a;
    int head_b, tail_b;
    int head_c, tail_c;
    void init() {
        head_a=tail_a=0;
        head_b=tail_b=0;
        head_c=tail_c=0;
    }
    void push(int v) {
        A[tail_a++] = v;
        // min-part
        while (head_b != tail_b && B[tail_b - 1] > v) tail_b--;
        B[tail_b++] = v;
        // max-part
        while (head_c != tail_c && C[tail_c - 1] < v) tail_c--;
        C[tail_c++] = v;
    }
    void pop() {
        if (head_a != tail_a) {
            if (A[head_a] == B[head_b]) ++head_b; // for min
            if (A[head_a] == C[head_c]) ++head_c; // for max
            ++head_a; // pop from main queue
        }
    }
    int get_min() {
        if (head_b == tail_b) return -1;
        return B[head_b];
    }
    int get_max() {
        if (head_c == tail_c) return -1;
        return C[head_c];
    }
    int Size() {
        return (tail_a-head_a);
    }
};

```